

# Spherical Environment Maps for Image-Based Rendering

Fausto Espinal  
Intelligent Systems Lab.  
Dept. of Computer Science  
University of South Carolina  
Columbia, SC 29208

Terry Huntsberger  
Intelligent Systems Lab.  
Dept. of Computer Science  
University of South Carolina  
Columbia, SC 29208

Peter R. Rogina  
Worldscape L.L.C.  
1 Waldron Drive  
Martinsville, NJ 08836

## ABSTRACT

Spherical environment maps capture all of the light visible from a given point in space. From such a representation it is possible to render any perspective view along any direction from a fixed eye position located at the center of the sphere. In this paper this idea is extended to allow for eye movement within a bounded region of space. A novel method for reconstructing views from a set of precaptured images is presented. Image information is manipulated to create new views of the captured environment. These views provide correct parallax information from the particular viewpoint and, therefore, enable occluded regions to be viewed. Unlike other related methods, there is no reliance on geometric information, thus making the approach completely content independent. Basic interpolation techniques are employed to improve the quality of the reconstructed images and results are presented.

**Keywords:** image-based rendering, computer graphics, telepresence, virtual reality, immersive environments

## 1. INTRODUCTION

In recent years much research has been undertaken in an effort to bypass the traditional graphics pipeline by directly modeling and encoding the light information at a scene. This information is contained in the *plenoptic function*<sup>1</sup> which is a five-dimensional quantity describing the flow of light at any position  $(x, y, z)$  and along any view direction  $\phi, \theta$ .

Quicktime VR<sup>2</sup> was the first to prove that it was possible to record, manipulate, and playback the light information at a scene. A subset of the plenoptic function is captured along a cylindrical environment map and replayed interactively to a user. McMillan<sup>3</sup> followed this by developing a novel method for registering cylindrical images which allows for reprojection of the pixels in these images to cylindrical images in other positions. He describes a method similar to Painter's algorithm which guarantees correct back to front pixel rendering order. One of the drawbacks to this method is that the optical flow must be calculated to obtain this registration which can sometimes be difficult.

Levoy et al.<sup>4</sup> and Gortler et al.<sup>5</sup> use cameras arranged on a rectangular grid to obtain a subset of the plenoptic function which they refer to as a Lightfield and a Lumigraph, respectively. In both works, inward-looking views of an object are reconstructed at interactive speeds. The Lumigraph uses depth correction techniques to obtain better registration in the final rendered image which means that geometric information must be obtained about the object being viewed. Of great importance in these works is the exploitation of texture mapping hardware to accelerate the playback speeds of the renderings. This is obtained by the rectangular arrangement of the data which allows for bilinear interpolation operations commonly implemented in texture mapping hardware systems.

The work described herein differs from previous research in several important areas. For the purposes of this paper, the light flow information is defined as incoming into a spherical region of space. A viewer observing the captured environment can look outwards in any direction with the freedom to move within a bounded region inside the sphere. This spherical arrangement represents a substantial departure from previously attempted work.

Geometric information of arbitrary scenery is very hard to obtain. This is especially true of real world scenery where the geometry is exceedingly complex and must therefore be assumed not to be available. While there is substantial work being done in the field of computer vision, the methods presented herein do not require any

---

Further author information:

Fausto J. Espinal: E-mail: espinal@cs.sc.edu, Tel. (803) 777-8308

Terry Huntsberger: E-mail: terry@cs.sc.edu, Tel. (803) 777-2404

Pete Rogina: E-mail: wrldscape@worldnet.att.net, Tel. (732) 764-0442

knowledge of the underlying geometry present in the environment being viewed. However, it is recognized that geometry information could potentially be used to affect reconstructed image quality in a manner similar to the Lumigraph.<sup>5</sup>

The remainder of this paper is organized as follows, first is a perspective about the relationship of this work to previous work done in the field of holography. A definition of spherical environment maps follows with information on how to build them using images captured with ordinary cameras. These ideas are then extended to allow for eye position movement and results are presented.

## 2. RELATIONSHIP TO HOLOGRAPHY

Much of the work in the field of image-based rendering is being done to simulate the type of seamless visual playback that has been demonstrable in the holographic arts for many years. Holographic stereograms offer the best example of this similarity in that they are created from a series of 2-D images and ultimately are capable of delivering a seamless representation of 3-D objects. Most holographic stereograms offer horizontal parallax only by supplying a successive series of vertical slit holograms.<sup>22–24</sup> Each slit provides a thin holographic window to a 2-D image located at an image plane. If the slits are thin and numerous enough for a set of viewing requirements, the seamless 3-D nature of the final product is assured. There are also fine examples of holographic stereography which employ matrices of holograms rather than slits.<sup>25</sup> This approach enables both horizontal and vertical parallax. Once again, if the matrix elements offer enough resolution, seamlessness is assured. It is this latter work, applied flexibly and digitally, that provides the closest analogy to today’s image-based rendering ambitions and challenges.

In the event that there are not enough images available to map into the matrix elements, interpolation techniques have been employed to generate intermediate parallax images which, when applied holographically, address the difficult problems associated with obscured surfaces.<sup>26</sup> More recently, Halle described a class of stereograms called Ultragrams which, among other things, model distortion characteristics with great accuracy.<sup>6,27</sup> Additional work in the field of interactive holographic display has been done by St. Hillaire and describes in great detail considerations for optimizing the capture and display of seamless 3-D scenes.<sup>28</sup>

The work presented herein strives to leverage the physics of holography with the necessary elements of computer science and mathematics to achieve what is quickly becoming a promising new immersive medium.

## 3. BUILDING SPHERICAL ENVIRONMENT MAPS

A spherical environment map can be thought of as all of the light information visible from a given point in space projected onto the inside surface of a sphere located concentrically surrounding that point. The information can be encoded onto a two-dimensional angular plane using the discretized form of the parametric equation of a sphere of radius  $R$ ,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} R \cdot \cos(\phi)\sin(\theta) \\ R \cdot \sin(\phi) \\ R \cdot \cos(\phi)\cos(\theta) \end{bmatrix}, \quad (\phi \in [-90, 90], \theta \in [-180, 180]). \quad (1)$$

This representation samples the poles of the sphere more densely and thus gives a non-uniform spatial distribution of points. The tradeoff is the ability to achieve an efficient representation from which one can quickly replay the content by warping portions of it onto perspective views of the surrounding scenery.

In order to build such a map, the environment must first be captured by taking pictures from the desired capture point along different directions. These images must then be warped onto a sphere along the corresponding directions. This can be achieved by sampling the sphere along all of the directions and calculating which pixels from the different images correspond to each direction. For a given position in the environment map corresponding to angular directions  $\phi_m, \theta_m$ , the address of the appropriate pixel can be calculated from the camera pointing along direction  $\phi_c, \theta_c$  by applying the following procedure.

First, bring the camera back to some reference coordinate system by applying the inverse rotation transform:

$$d' = R_{\phi_c, \theta_c}^{-1} \cdot d_{\phi_m, \theta_m}, \quad (2)$$

where  $d_{\phi_m, \theta_m} \in \mathbb{R}^3$  is the cartesian representation of the unit vector with directions  $\phi_m$  and  $\theta_m$  and  $R_{\phi_c, \theta_c}^{-1} \in \mathbb{R}^{3 \times 3}$  is the rotation matrix. Once this inverse rotation transformation is performed, the given camera direction is aligned with the reference camera direction.

Then calculations to determine the row and column information in the image are performed using the formulas:

$$r = d'_y \left[ \frac{F}{d'_z} \right] + \frac{H}{2} \quad (3)$$

and

$$c = d'_x \left[ \frac{F}{d'_z} \right] + \frac{W}{2}, \quad (4)$$

where  $F$  is the camera focal length and  $H$  and  $W$  are the height and width of the camera image plane. Using this procedure, pixels from different cameras will map to the same location in the environment map. This is advantageous in reducing noise and aliasing from the map since these values can be summed and averaged to produce one value. Portions of this map are then warped onto perspective views of desired directions and fields of view. Figure 1 shows a spherical environment map of a conference room which was obtained using the method described above. The rows represent change in the pitch angle  $\phi_m$ , and the columns represent change along the yaw angle of the sphere,  $\theta_m$ .



**Figure 1.** *Spherical environment map of a conference room. The rows represent change along the pitch angle  $\phi_m$  and the columns represent change along the yaw angle of the sphere  $\theta_m$ .*

An interactive viewer, which permits a user to look around a point at image resolutions of  $256 \times 256$  has been built. This viewer will reconstruct views from the given point in space where the map was captured. Once the viewer was complete, the focus turned to the more challenging problem of enabling view reconstruction from within a defined spatial locus surrounding the center point of the sphere. This is the key to ultimately providing content-independent immersive viewing of environments.

The issue of surface occlusion has historically been difficult to overcome. Surface occlusion is not, however, a problem in holography since the medium allows for light rays to be directed in such a way as to have adequate dimensional flexibility to achieve seamless content independence. That is, the medium is inherently multi-dimensional to an adequate degree for enabling seamless reconstruction over a locus of viewing locations.

A method capable of reconstructing views from a locus surrounding a point is now presented.

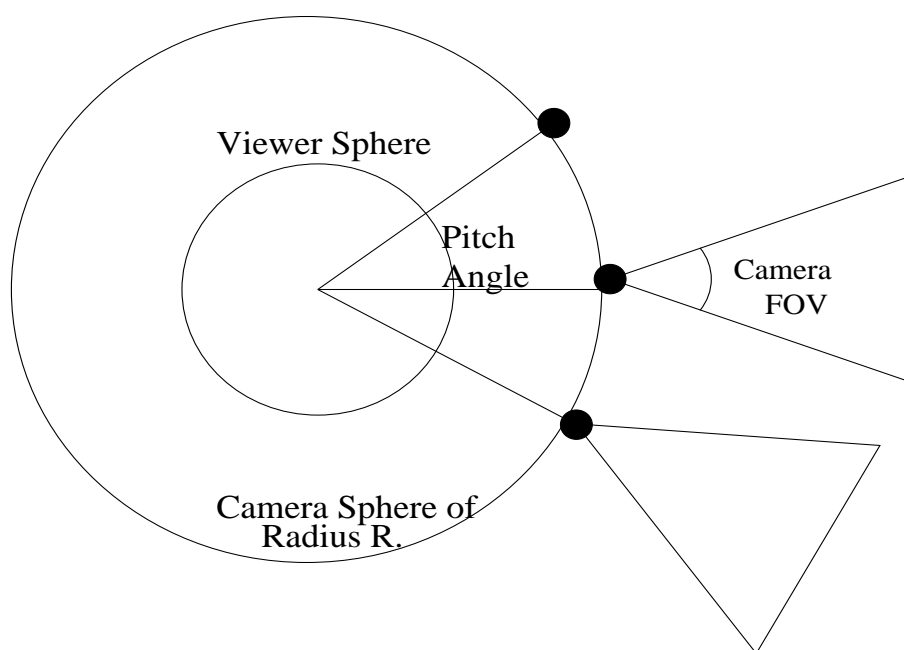
#### 4. ALLOWING FOR EYE-POSITION MOVEMENT

The objective of this work is to extend the idea of a spherical environment map to allow for eye position movement within a bounded locus inside a sphere. A method of efficiently capturing and encoding the light flowing into a spherical region of space was needed. Specifically, the following tasks were defined:

- Sample the light information inside a spherical region of space using a calibrated camera with known field of view and focal distance.
- Allow for a range of movement of about 18 inches in real world terms. This will be shown later to be a function of the number of images taken and the field of view of the capture camera.

A camera field of view of 60 degrees was chosen because it is relatively easy to find a very linear lens with this field of view and we wished to avoid the extra complexities introduced with fisheye lens transfer functions.

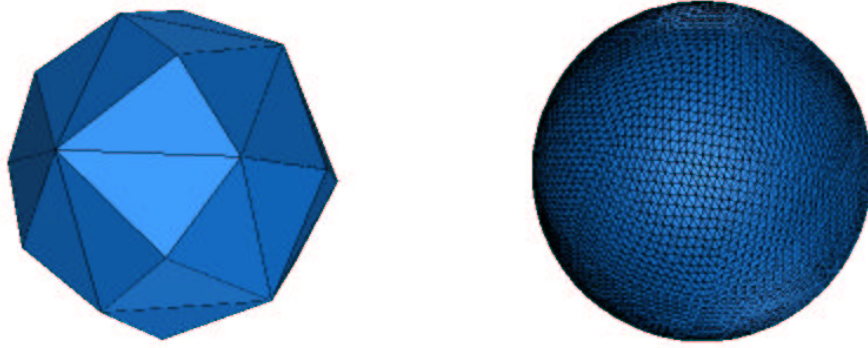
Images were rendered by pointing the virtual camera radially outwards along the surface of a sphere (referred to as the capture sphere) of known radius  $R$ . The capture sphere encloses a smaller region of space from which any view of the scene can be reconstructed (referred to as the viewpoint locus). Figure 2 shows a side view of the capture scheme.



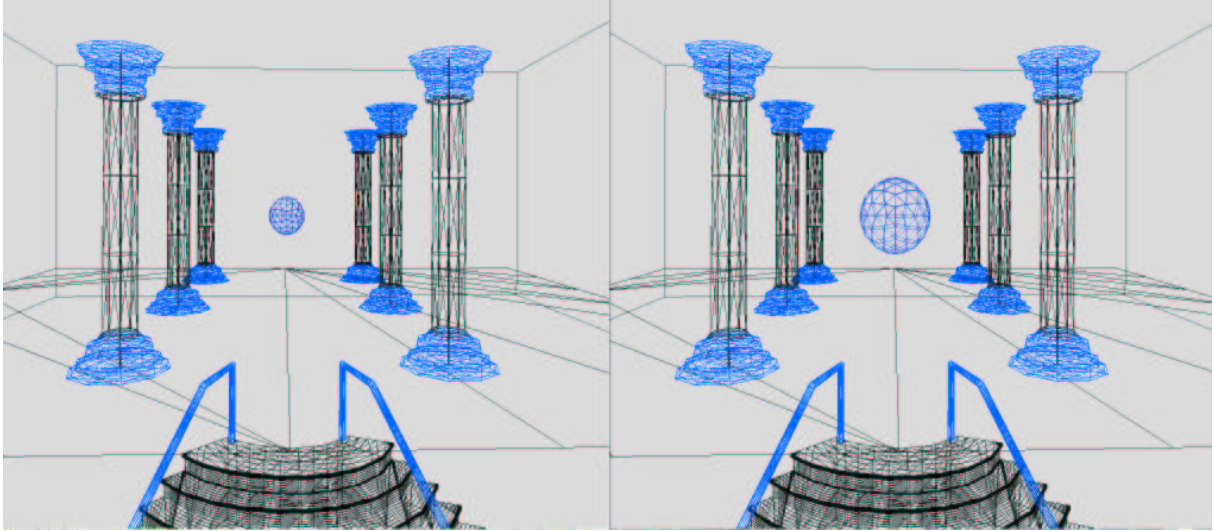
**Figure 2.** Side view of the viewer locus embedded inside the capture sphere.

The first question encountered was how to sample the sphere such that all the gaps were covered. A standard spherical discretization creates a large concentration of points near the poles of the sphere and thus a highly non-uniform spatial distribution of cameras. Instead, platonic solids, commonly used in computer graphics as approximations of spheres, were chosen. This results in a more complex addressing scheme when compared to the angular discretization method, but produces a much smaller number of cameras and more uniform spatial coverage. Figure 3 shows the base polyhedra and its extrusion which was used for our environment capture.

The next consideration was how large the capture sphere should be. Figure 4 shows two examples of capture spheres within a computer generated environment. The radius of this sphere is related to the number of cameras chosen for the capture and their field of view. If the radius of the sphere is increased, more cameras will be needed to sample its larger surface area or the field of view of the cameras would have to be increased. A larger value of the radius also makes the capture sphere approach the objects in the scene and has the additional effect of increasing the volume of the viewpoint locus, thus providing more potential parallax motion.



**Figure 3.** The first figure shows the base polyhedra used for our camera sampling. The second figure shows the same polyhedra after four levels of extrusion. This produced 6146 vertices and a uniform sampling of the sphere.

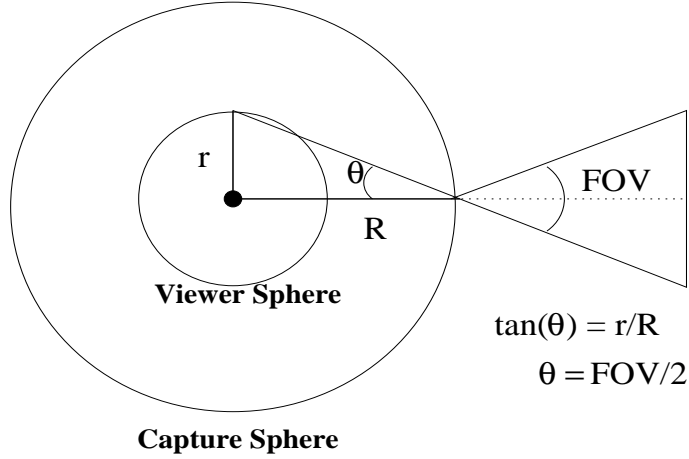


**Figure 4.** Two examples of capture spheres within an environment surrounded by pillars. The size of the capture sphere and the field of view determine the number of cameras which will cover particular regions of space. Sphere size and field of view also affect the region inside which the viewer is constrained.

The next step was to determine the bounds of the viewpoint locus. This is a geometric problem. The regional constraint arises because at some point the viewer can be looking towards a region of the environment for which there is no coverage provided by any camera on the capture sphere. This will generally occur when reconstructing a wide field of view at points nearer the capture sphere and looking radially outward. The viewpoint locus is affected by the camera field of view, the reconstruction field of view, the orientation of the reconstructed image (i.e., the tilt), and the viewing direction (e.g., looking straight outward or back through the center). For the purposes of this paper, the variables were limited by not incorporating tilt to the reconstructed images and only modeling the case of looking outward from a positive  $r$  value, where  $0 \leq r \leq R$ . It is possible to solve for  $r$  by using the similar triangle rule as illustrated in Figure 5 and in the following formula :

$$r = R \cdot \tan\left(\frac{FOV}{2}\right), \quad (5)$$

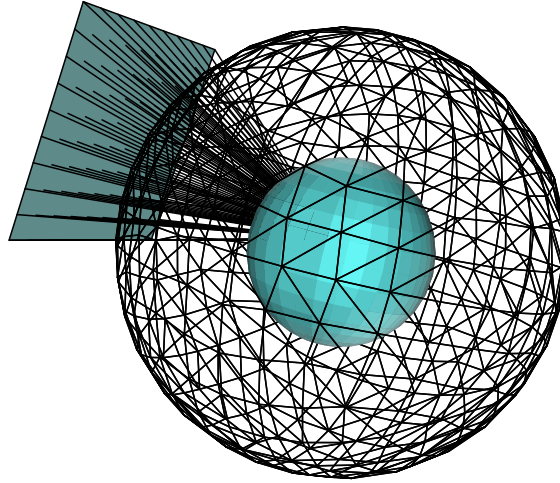
where  $R$  is the radius of the capture sphere and  $FOV$  is the field of view of the capture cameras.



**Figure 5.** *Determination of a bound for the viewpoint locus. The field of view of the cameras determine the maximum distance a viewer can move without exceeding the available scene information.*

## 5. FRAME RECONSTRUCTION

Arbitrary views from any point within the viewpoint locus can be reconstructed in a straight forward way. Figure 6 shows a sample view frustum to be reconstructed and its relationship to the capture sphere and a sample viewpoint locus.



**Figure 6.** *View frustum to be reconstructed. The capture sphere containing a viewpoint locus from which a view frustum to be reconstructed emanates.*

For clarity, the following terms are defined. Define  $P = (p_x, p_y, p_z)$  to be the eye position vector and  $D = (d_x, d_y, d_z)$  to be the view frustum direction vector.  $R$  and  $r$  are the radii of the capture sphere and the viewpoint

locus respectively.  $VF$  represents the set of rays  $V_i = (P + D_i t)$  composing a view frustum. The general procedure to reconstruct a given view frustum is described by the following pseudo-code:

**Procedure ReconstructFrame**( $V = P + Dt$ )

$V, P, D \in \mathbb{R}^3$

1.  $VF = \text{BuildViewFrustum}(V)$ . Calculate the set of points composing the viewpoint frustum image plane. This can be done quickly since the points are coplanar and the origin is  $P$  for all the rays.
2. ForEach  $((V_i = (P + D_i t)) \in VF)$ 
  - Calculate the intersection point  $S_i$  of  $V_i$  with the capture sphere ( $\|S_i\| = R$ ).
  - Find the  $n$  nearest cameras to  $S_i$ .
  - Calculate intersection pixel for each of the  $n$  nearest cameras using equations 2, 3 and 4.
  - Interpolate pixel value  $p_i$  from the  $n$  pixel values. If  $n = 1$  then, simply use a nearest camera scheme.
  - Place this pixel value  $p_i$  at the position in the reconstructed image corresponding to ray  $V_i$ .
3. Output desired view frame.

The step of calculating the intersection pixel for the  $n$  nearest cameras has a built-in inherent inaccuracy. This is due to the fact that the reconstructed view ray is being projected from the cameras on the capture sphere instead of from the true location co-ordinate inside the viewpoint locus. This will manifest itself in the reconstructed images as aliasing artifacts. Figure 7 is an exaggerated illustration of this effect showing how the incorrect pixel value is picked up from the nearest camera image instead of from the reconstruction viewpoint. The effects of this phenomenon are minimized by adequate camera coverage on the capture sphere. This will serve to minimize the translation errors across the sphere.

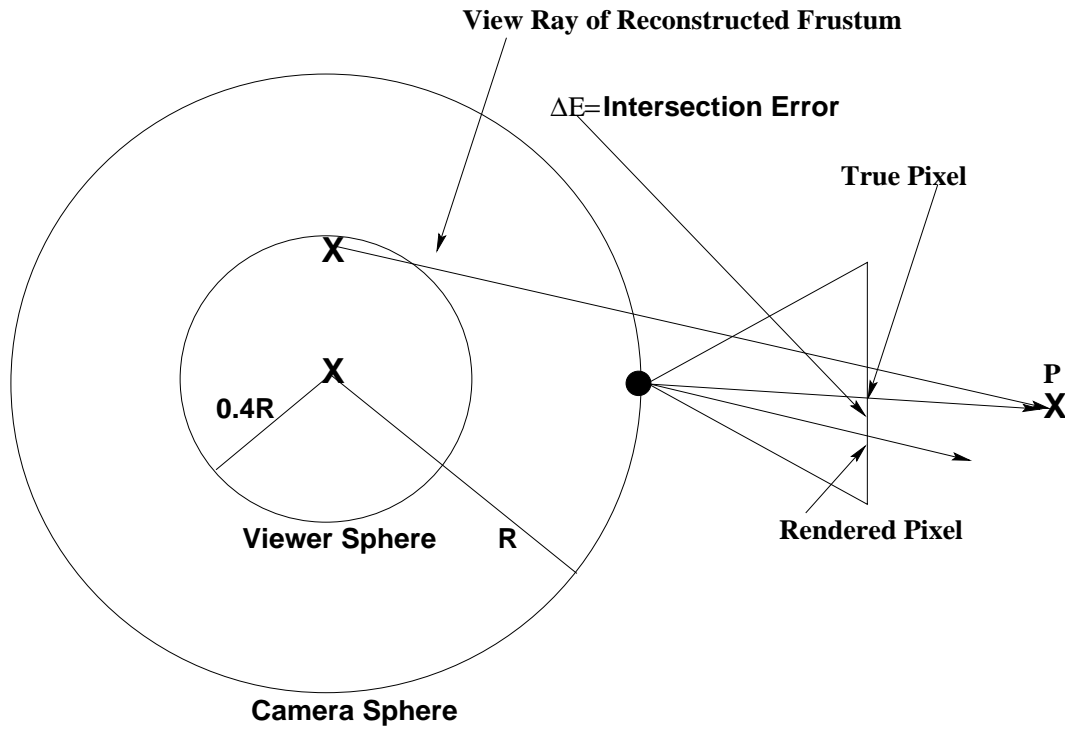
Other factors affecting the magnitude of this aliasing error are: 1) the distance of the capture sphere to the objects in the environment, and 2) the radius of the capture sphere.

The closer the objects in the environment are to the cameras, the more motion parallax will be present in neighboring frames. This will in turn cause the ray-plane intersection error, described above, to be more apparent. The radius of the capture sphere also affects the motion parallax present in the neighboring frames as well as affecting the volume of the viewpoint locus. Therefore, these parameters need to be balanced appropriately to minimize aliasing. These considerations will also affect the apparent scale of the environment. For this work, dimensions have been selected on the basis of experimentation with the goal of simulating a life-size reconstructed scene.

## 6. RESULTS

In the presentation of the frame reconstruction procedure, interpolation is done between the nearest  $n$  cameras to the intersection point on the capture sphere. Figures 8, 9 and 11 show the interior of a U.S. Navy cruiser. Figure 8 shows two reconstructed images of this artificial environment with values of  $n = 1$  and  $n = 3$  respectively. The right image is smoother and some of the aliasing artifacts have been removed along the borders of the walls and other objects in the scene.

Figures 9 and 10 show results of side-to-side motion parallax achievable using these methods. In figure 9, notice the portion behind the right wall that comes into view and the view of the door as the eye position is moved horizontally from left to right. Figure 10 was reconstructed from a public domain 3-D scene and was chosen for calibration purposes. The motion parallax is more visible in this figure, although the aliasing artifacts are more prevalent in the pillar due to its nearness to the capture sphere and the stressful optical pattern mapped onto it. It is important to note that an appropriate choice of texture for the pillar would make these artifacts much less noticeable. Certain marbled texture maps made the aliasing virtually undetectable. Movies of both of these environment can be obtained at <http://www.cs.sc.edu/~espinal/ibr.htm>.



**Figure 7.** *Exaggerated illustration of ray-plane intersection error.*

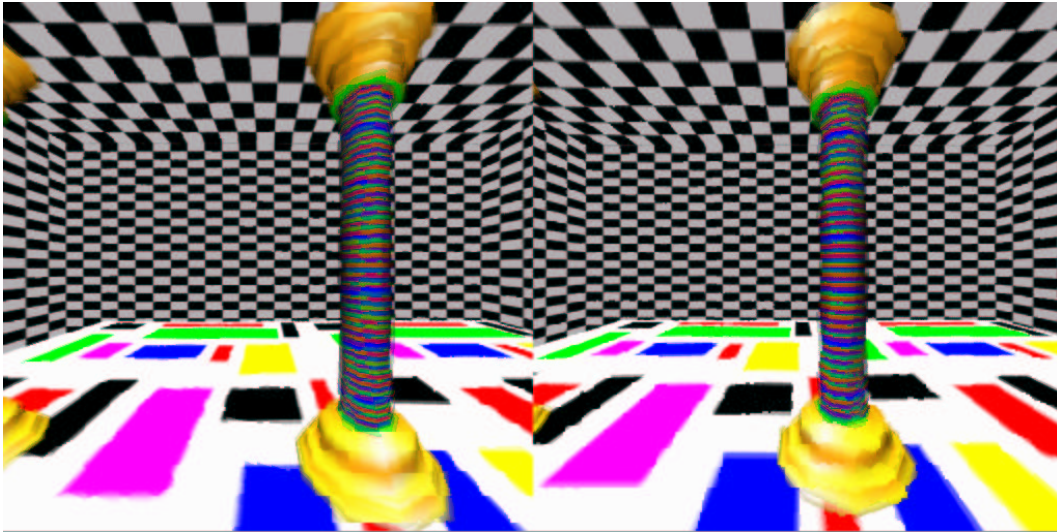


**Figure 8.** *A non-interpolated frame ( $n = 1$ ) versus an interpolated frame ( $n = 3$ ). The picture on the left is obtained by picking the nearest camera's pixel value. The picture on the right is obtained by averaging the vertices on the triangular face of the polyhedral figure through which the view rays pass.*

Figure 11 shows another example of the motion parallax which can be achieved as the viewer moves from right to left. It also shows noticeable aliasing along the railing. This is again due to the short distance between the capture sphere and the stairway of the cruiser which increases the ray-plane intersection error.



**Figure 9.** *A sample view of the motion parallax. Two frames rendered from different horizontal positions within the viewer locus.*



**Figure 10.** *Another sample view of motion parallax. Two frames rendered from different positions within the viewer locus.*

## 7. CONCLUSIONS AND FUTURE WORK

It has been shown that it is possible to capture and redisplay the light information visible from a spherical region of space by taking pictures of a surrounding environment along a sphere. Due to the large amounts of image data generated by our methods, much work has yet to be done in efficiently compressing and decompressing this information. Our preliminary studies have shown there is a great deal of correlation that can be exploited in these data sets which should make high compression ratios achievable. Areas of continuing work include achieving very fast frame reconstruction so as to playback an environment immersively in real-time.

## ACKNOWLEDGMENTS

This work was done under a grant from WorldScape L.L.C. and also supported by ONR Grant N00014-97-1-1163.

The authors wish to thank everybody who provided useful comments and criticisms for the completion of this work. We would specifically like to acknowledge the work of Sreenadha Godavarthy for the implementation of the



**Figure 11.** Two frames rendered from two different positions within the viewpoint locus, as the eye position is moved from right to left.

frustum-sphere intersection algorithm; Rajesh Chandran for the Java port of the interactive spherical environment map viewer and Mike Ehrlich for reviewing the reconstruction algorithms.

We would also like to give credit to the creators of 3-D scenes used in this work. The pillars data set in figure 10 was taken from the Avalon public domain ftp site. The cruiser data set was taken from the radiance ftp site and was created by Saba Rofchaei and Greg Ward. The conference room data set was also taken from the radiance ftp site and was created by Anat Grynberg and Greg Ward. The availability of these data sets saved us a great deal of time in obtaining our results.

## REFERENCES

1. Edward H. Adelson and James R. Bergen. The plenoptic function and the elements of early vision. *Computational Models of Visual Processing*, pages 3–20, 1991.
2. Shenchang Eric Chen. Quicktime vr - an image-based approach to virtual environment navigation. In *Proceedings of SIGGRAPH 95*. SIGGRAPH 95, 1995.
3. Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *Proceedings of SIGGRAPH 95*, pages 1–8. SIGGRAPH 95, 1995.
4. Mark Levoy and Pat Hanrahan. Light field rendering. In *Computer Graphics Proceedings*, pages 31–42. SIGGRAPH 96, 1996.
5. Richard Szeliski Steven J. Gortler, Radek Grzeszczuk and Michael F. Cohen. The lumigraph. In *Computer Graphics Proceedings*, pages 43–54. SIGGRAPH 96, 1996.
6. Michael A. Klug Michael W. Halle, Stephen A. Benton and John S. Underkoffler. The ultragram: A generalized holographic stereogram. *SPIE Proceeding Practical Holography V*, 1991.
7. Steven M. Seitz and Charles R. Dyer. View morphing. In *Proceedings of SIGGRAPH 96*, pages 21–30. SIGGRAPH 96, 1996.
8. Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of SIGGRAPH 97*, pages 251–258. SIGGRAPH 97, 1997.
9. Goerge Wolberg. *Digital Image Warping*. IEEE Computer Society Press Monograph, 1990.
10. Li-Yi Wei. Light field compression using wavelet transform and vector quantization. Article obtained from web page., 1997.
11. John Vince. *Virtual Reality Systems*. ACM Press, 1995.
12. Michael F. Cohen Steven J. Gortler, Li-wei He. Rendering layered depth images. Technical report, Microsoft Research, 1997.

13. Leonard McMillan. Acquiring immersive virtual environments with an uncalibrated camera. Technical Report 95-006, University of North Carolina, 1995.
14. Leonard McMillan. A list-priority rendering algorithm for redisplaying projected surfaces. Technical Report 95-005, University of North Carolina, 1995.
15. Ned Greene. Environment mapping and other applications of world projections. *IEEE Computer Graphics and Application*, 1986.
16. Richard I. Hartley. Self-calibration from multiple views with a rotating camera. In *Proceedings of the European Conference on Computer Vision*, pages 471–478. ECCV 94, 1994.
17. M. Dhome N. Daucher and J.T. Lapreste. Camera calibration from spheres images. In *Proceedings of the European Conference on Computer Vision*, pages 449–454. ECCV 94, 1994.
18. Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4), august 1986.
19. Ken-Ichi Kanatani. Transformation of optical flow by camera rotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2), march 1988.
20. Paul W. Jones Majid Rabbani. *Digital Image Compression Techniques*. SPIE-The International Society for Optical Engineering, 1991.
21. Khalid Sayood. *Introduction to Data Compression*. Morgan Kaufmann Publishers, Inc., 1996.
22. J. D. Redman. Three dimensional reconstruction of people and outdoor scenes using holographic multiplexing. *J. Sci. Inst.*, 8(2):1740, 1969.
23. M. C. King, A. M. Noll, and D. H. Berry. A new approach to computer generated holography. *Applied Optics*, 9-12:471–475, February 1970.
24. D. J. De Bitetto. Holographic panoramic stereograms synthesized from white light recordings. *Applied Optics*, 8-8:1740–1741, August 1969.
25. John R. Andrews and Michael D. Rainsdon. Image processing for animated holograms. In *Practical Holography VI*, volume 1667, pages 120–126. SPIE, 1994.
26. S. Takahashi, T. Honda, M. Yamaguchi, N. Ohyama, and F. Iwata. Generation of intermediate parallax-images for holographic stereograms. In *Practical Holography VII*, volume 1914. SPIE, 1993.
27. M. W. Halle. The generalized holographic stereogram. Master’s thesis, Massachusetts Institute of Technology, 1991.
28. Pierre St Hilaire. Modulation transfer function of holographic stereogram. In *Applications of Optical Holography*. SPIE, 1994.